CLAIMS

I claim:

1. A method of monitoring and controlling instruction dependency for microprocessors, the method comprising:

fetching an instruction at a thread control element;

comparing one or more source operand identifications of the instruction to one or more temporary register identifications, wherein each of the one or more temporary register identifications is stored in a temporary register identification pipeline of a set of one or more temporary register identification pipelines; and

verifying whether any of the one or more source operand identifications matches any of the one or more temporary register identifications.

2. The method of Claim 1, wherein none of the one or more source operand identifications matches any of the one or more temporary register identifications.

3. The method of Claim 2, further comprising the step of initiating execution of the instruction.

4. The method of Claim 3, further comprising the step of verifying whether a destination operand of the instruction is a temporary register.

5. The method of Claim 4, wherein the destination operand is not a temporary register.

6. The method of Claim 5, further comprising the step of writing a null value into a first pipeline of the set of one or more temporary register pipelines.

7. The method of Claim 4, wherein the destination operand is a temporary register.

8. The method of Claim 7, further comprising the step of writing an identification corresponding to the destination operand into a first pipeline of the set of one or more temporary register pipelines.

9. The method of Claim 1, wherein the content in all except the last of the set of one or more temporary register pipelines is shifted to the next pipeline at the beginning of each clock cycle.

10. The method of Claim 9, wherein the content of the last pipeline of the set of one or more temporary register pipelines is released at the beginning of each clock cycle.

11. The method of Claim 1, wherein at least one of the one or more source operand identifications matches one of the one or more temporary register identifications.

12. The method of Claim 11, further comprising the step of prohibiting execution of the instruction.

13. The method of Claim 12, further comprising the step of comparing the one or more source operand identifications to the one or more temporary register identifications at the beginning of each clock cycle until none of the one or more source operand identifications matches any of the one or more temporary register identifications.

14. The method of Claim 13, further comprising the step of verifying whether a destination operand of the instruction is a temporary register.

15. The method of Claim 14, wherein the destination operand is not a temporary register.

16. The method of Claim 15, further comprising the step of writing a null value into a first pipeline of the set of one or more temporary register pipelines.

17. The method of Claim 14, wherein the destination operand is a temporary register.

18. The method of Claim 17, further comprising the step of writing an identification corresponding to the destination operand into a first pipeline of the set of one or more temporary register pipelines.

19. A method of monitoring and controlling instruction dependency for microprocessor systems, the method comprising:

a) fetching an instruction at a thread control element;

b) receiving an instruction request at an arbiter, wherein the instruction request is issued from the thread control element;

c) comparing one or more source operand identifications of the instruction to one or more temporary register identifications, wherein each of the one or more temporary register identifications is stored in a temporary register identification pipeline of a set of one or more temporary register identification pipelines;

d) verifying whether any of the one or more source operand identifications matches any of the one or more temporary register identifications; and

e) if none of the one or more source operand identifications matches any of the one or more temporary register identifications:

e1) verifying whether a destination operand of the instruction is a temporary register; and

e2) if the destination operand of the instruction is a temporary register:

writing an identification corresponding to the destination operand into a first pipeline of the set of one or more temporary register pipelines.

20. The method of Claim 19, further comprising the step of initiating execution of the instruction.

21. The method of Claim 19, if the destination operand of the instruction is not a temporary register in step e2), further comprising the step of writing a null value into a first pipeline of the set of one or more temporary register pipelines.

22. The method of Claim 19, if at least one of the one or more source operand identifications matches one of the one or more temporary register identifications in step e), further comprising the steps of:

prohibiting the execution of the instruction;

reiterating step d) until none of the one or more source operand identifications matches any of the one or more temporary register identifications; and

verifying whether a destination operand of the instruction is a temporary register.

23. The method of Claim 22, wherein the destination operand is a temporary register.

24. The method of Claim 23, further comprising the step of writing an identification corresponding to the destination operand into a first pipeline of the set of one or more temporary register pipelines.

25. The method of Claim 22, wherein the destination operand is not a temporary register.

26. The method of Claim 25, further comprising the step of writing a null value into a first pipeline of the set of one or more temporary register pipelines.

27. The method of Claim 19, wherein the content in all except the last of the set of one or more temporary register pipelines is shifted to the next pipeline at the beginning of each clock cycle.

28. The method of Claim 27, wherein the content of the last pipeline of the set of one or more temporary register pipelines is released at the beginning of each clock cycle.

29. A system for instruction dependency monitor and control, comprising:

a set of one or more thread control elements for fetching instructions;

a set of one or more comparing elements, wherein each of the one or more comparing elements is coupled to a corresponding thread control element in the set of one or more thread control elements; and

a set of one or more temporary register identification pipelines, wherein the one or more temporary register identification pipelines are coupled to the one or more comparing elements.

30. The system of Claim 29, further comprising an instruction buffer coupled to the one or more thread control elements.

31. The system of Claim 30, further comprising an arbiter, wherein the arbiter is coupled to the one or more thread control elements, the one or more comparing elements, and the one or more temporary register identification pipelines.

32. The system of Claim 31, further comprising an arithmetic logic unit (ALU) coupled to the arbiter.

33. The system of Claim 32, further comprising a set of one or more input data buffers coupled to the arbiter, wherein each input data buffer corresponds to a thread control element of the one or more thread control elements.

34. The system of Claim 33, further comprising a set of one or more temporary register buffers coupled to the arbiter, wherein each temporary register buffer corresponds to a thread control elements of the one or more thread control elements.

35. A system for instruction dependency monitor and control, comprising:

a set of one or more thread control elements for fetching instructions;

a set of one or more comparing elements, wherein each of the one or more comparing elements is coupled to a corresponding thread control element in the set of one or more thread control elements;

a set of one or more temporary register identification pipelines, wherein the one or more temporary register pipelines are coupled to the one or more comparing elements, and

an arbiter coupled to the thread control elements, the comparing elements, and the temporary register pipelines.